

1. (Currently Amended) A method of executing a computer program distributed across a plurality of computers, said method comprising ~~the steps of~~:

a) obtaining available excess computer capacity from a plurality of potential ~~applicants~~

participants, wherein each participant registers at least one computer by obtaining a committed number of hours for said computer and determining a normalized excess capacity for said computer;

b) partitioning a computer program into a plurality of independent tasks of approximately equal size;

c) distributing said tasks to said participants according to available excess capacity;

d) determining whether each distributed task will execute within a selected range of other said distributed tasks;

e) beginning execution of said distributed tasks;

f) receiving completed tasks from said participants; and

g) determining whether every task has been executed by at least one participant.

2-8. (Canceled)

9. (Currently Amended) A method ~~of doing business~~ as in claim 8 1, wherein each ~~new~~ said ~~potential~~ participant is provided with one or more benchmark tasks, said ~~new~~ participants' normalized excess capacity being adjusted responsive to performance of said ~~one or more~~ benchmark tasks.

10. (Currently Amended) A method as in claim 1, wherein ~~said in step (b)~~ of partitioning of the computer program includes assigning a plurality of said independent tasks from said partition are assigned to a plurality of registered computers participating machines in an overlapping tiling manner that minimizes dependence on individual computers.

11-13. (Canceled)

14. (Currently Amended) A method as in claim 1, wherein the ~~step (d)~~ of determining whether each task will execute within the selected range further includes reassigning any task determined to be unlikely to execute within said range.

15. (Currently Amended) A method as in claim 1, wherein as each completed task is received ~~in step (f)~~, a check is made to determine whether said completed task is on schedule.

16. (Original) A method as in claim 10, wherein any participant producing a task that is not on schedule is determined to have a slow machine and other tasks assigned to such slow machines are reassigned to other available participants.

17. (Canceled)

18. (Currently Amended) A distributed processing system for transferring excess capacity from a plurality of computers to a party requiring execution of a computer program, said distributed processing system comprising:

a plurality of participating computers connected together over a network, each said computer

being registered as available for a committed number of hours;

means for determining a normalized excess capacity for each participating computer;

means for partitioning a computer program into a plurality of independent tasks of

approximately equal size;

means for distributing said tasks to said participating computers according to normalized excess capacity;

means for determining whether each distributed task will complete within a selected range of

other said distributed tasks and redistributing any of said tasks determined likely to

not complete within said selected range;

means for receiving completed tasks from said computers; and

means for determining whether each task has been executed by at least one computer.

19-26. (Canceled)

27. (Currently Amended) A computer program product for selling unused excess capacity of a plurality of connected computers to a party requiring execution of a partitionable computer program, said computer program product comprising a computer useable medium having computer readable program code thereon, said computer readable program code comprising:

computer readable program code means for registering a plurality of participating computers
as available for a committed number of hours;

computer readable program code means for partitioning a computer program into a plurality
of independent tasks of approximately equal size;

computer readable program code means for distributing said tasks to said registered
participating ~~computer~~ computers according to normalized excess capacity;

computer readable program code means for determining whether each distributed task will
complete within a selected range of other said distributed tasks and redistributing any
of said tasks identified as not completing within said selected range;

computer readable program code means for receiving completed tasks from said computers;
and

computer readable program code means for determining whether each task has been executed
by at least one computer.

28-33. (Canceled)